Assessment Report
on

"For these problems, you have to generate heat maps of confusion matrices and calculate the evaluation metrics such as accuracy, precision, recall for classification type problem and for other perform segmentation and clustering. Heart Rate Predictor"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

SHUBHAM KUMAR (20240110300244, CSE-AI D)

Under the supervision of

"MR.ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Heart disease is one of the leading causes of death worldwide. Early prediction can help in timely intervention and treatment. This project aims to develop a machine learning model that predicts whether a patient has heart disease based on several medical attributes.

Methodology

1. Objective

To classify whether a patient is likely to have heart disease using various medical parameters by:

Training a classification model

Evaluating its performance using metrics such as accuracy, precision, recall, and F1-score

Visualizing results using a confusion matrix heatmap

2. Dataset Description

The dataset contains anonymized patient records with 14 attributes:

Feature	Description
age	Age of the patient
sex	Sex (1 = male; 0 = female)
cp	Chest pain type (0–3)
trestbps	Resting blood pressure
chol	Serum cholesterol
fbs	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
restecg	Resting ECG results
thalach	Maximum heart rate achieved
exang	Exercise-induced angina
oldpeak	ST depression induced by exercise
slope	Slope of the ST segment
ca	Number of major vessels (0–3)
thal	Thalassemia (1 = normal, 2 = fixed defect, 3 = reversible defect)
target	1 = presence of heart disease, 0 = absence

---

3. Data Preprocessing

Since the original dataset was extracted from a PDF, some formatting issues were present:

Rows were merged or split incorrectly

Values appeared in string blocks

To fix this:

The script restructured the data using regex and string manipulation

Converted all columns to numeric types

Handled missing or malformed rows

---

4. Model Used

Algorithm: Random Forest Classifier

Reason: Robust to outliers, handles both categorical and continuous variables well, provides feature importance

---

5. Model Evaluation

The dataset was split into:

80% training data

20% testing data

The following metrics were calculated:

Confusion Matrix: Shows correct and incorrect predictions

Accuracy: (TP + TN) / Total

Precision: TP / (TP + FP)

Recall: TP / (TP + FN)

F1 Score: Harmonic mean of precision and recall

A heatmap of the confusion matrix was also generated to visualize model performance.

Code

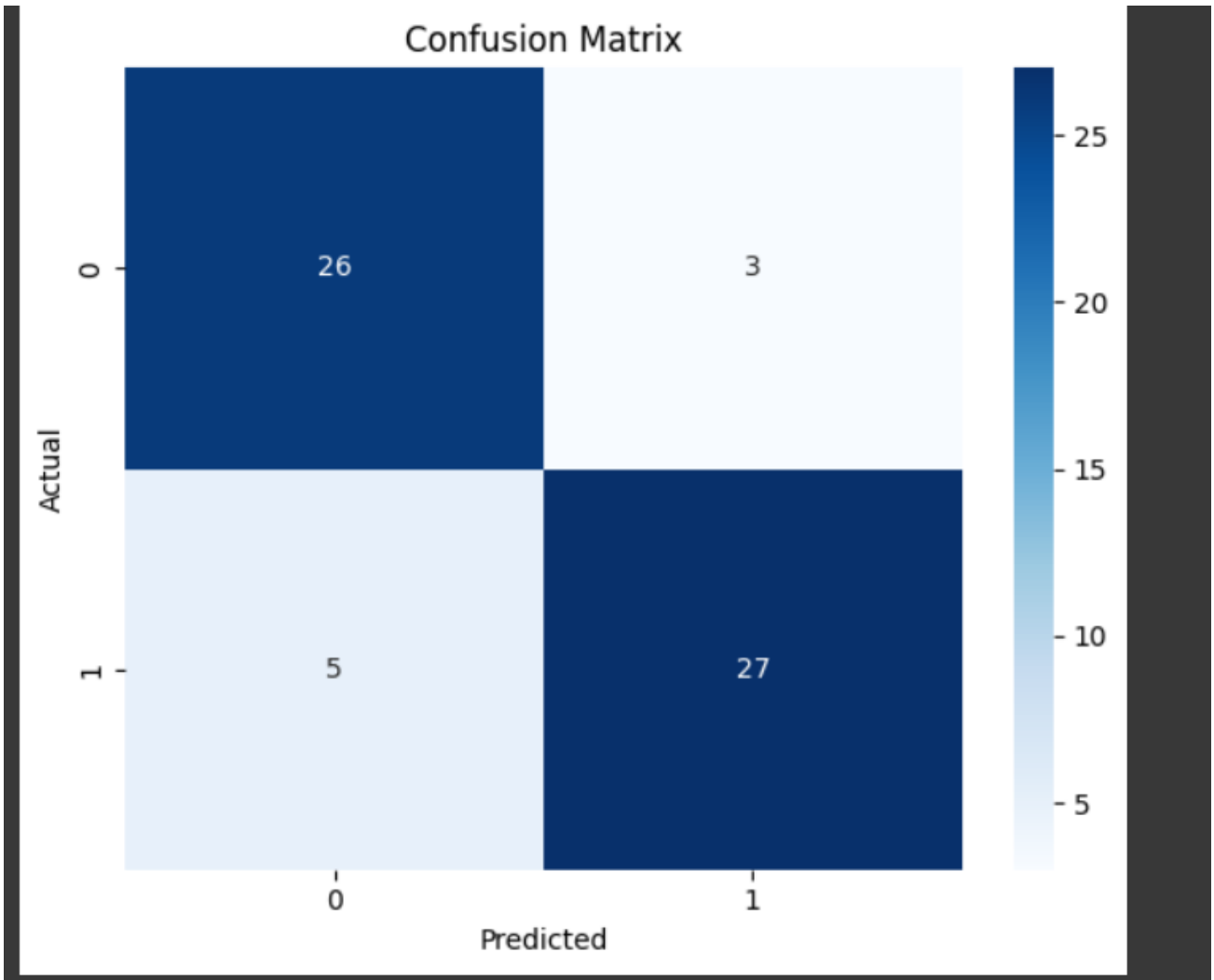
```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import (

    confusion_matrix, accuracy_score, precision_score,

    recall_score, f1_score

)



# === Step 1: Load and Clean the Dataset ===

# If the CSV has corrupted columns due to PDF conversion, fix it here

def load_and_fix_csv(path):

    raw_df = pd.read_csv(path)

    fixed_rows = []



    for i in range(len(raw_df)):
```

```
        row = [str(val) for val in raw_df.iloc[i] if pd.notna(val)]
        merged = ' '.join(row).split()
        if len(merged) == 14:
            fixed_rows.append(merged)

    columns = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
               'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target']

    df = pd.DataFrame(fixed_rows, columns=columns)
    return df.apply(pd.to_numeric)

# Replace this with your path
df = load_and_fix_csv("heart_rate.csv")

# === Step 2: Prepare Features and Target ===
X = df.drop("target", axis=1)
y = df["target"]

# === Step 3: Split and Train ===
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = RandomForestClassifier(random_state=42)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)



# === Step 4: Confusion Matrix Heatmap ===

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 5))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.tight_layout()

plt.show()



# === Step 5: Evaluation Metrics ===

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)
```

```
f1 = f1_score(y_test, y_pred)



print(f"Accuracy : {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall   : {recall:.2f}")

print(f"F1 Score : {f1:.2f}")
```

Output/Result




```
Accuracy : 0.87
Precision: 0.90
Recall   : 0.84
F1 Score : 0.87
```

References/Credits

UCI Machine Learning Repository: Heart Disease Dataset

Scikit-learn Documentation

Matplotlib & Seaborn for Visualization